

بسمه تعالی



تمرین سری اول درس یادگیری عمیق

پوریا محمدی نسب

۴۰۰۷۲۲۱۳۸

سوال اول

موارد زیر را از کتاب مرجع اصلی مطالعه فرمایید و سپس با زبانی ساده و به شکلی شفاف آنها را بیان نمایید.

1) Learning algorithm:

“The set of all distinct programs (input-output mappings) that we can produce just by manipulating the parameters is called a family of models. And the meta-program that uses our dataset to choose the parameters is called a learning algorithm.”

به مجموعه ای از برنامه ها (توابع) متمایز که ورودی را به خروجی نگاشت میکنند و ما میتوانیم تنها با تغییر دادن پارامترها آنها را تولید کنیم، خانواده ای از مدل ها گفته میشود. و به "ابر برنامه ای" که دیتاست را دریافت میکند و پارامترهای تولید تابع را تنظیم (انتخاب) میکند یک الگوریتم یادگیری گفته میشود.

2) Objective function:

“An objective function that quantifies how well (or badly) the model is doing.”

یک تابع هدف (objective function) به صورت کمی مشخص میکند که مدل چقدر خوب یا بد عمل میکند.

3) 4D tensors vs. 4-dimensional vector:

“Tensors give us a generic way of describing n-dimensional arrays with an arbitrary number of axes. Vectors, for example, are first-order tensors.”

تانسور یک راه کلی توصیف آرایه های n بعدی است برای مثال 4D tensor یعنی دیتای ما به حالت 4 بعدی در نظر گرفته میشود اما dimensionality در وکتورها به معنی تعداد اعضای آن وکتور است. برای مثال به وکتوری که 4 عدد حقیقی در آن ذخیره شده باشد یک 4-dimensional vector است.

4) Reinforcement Learning:

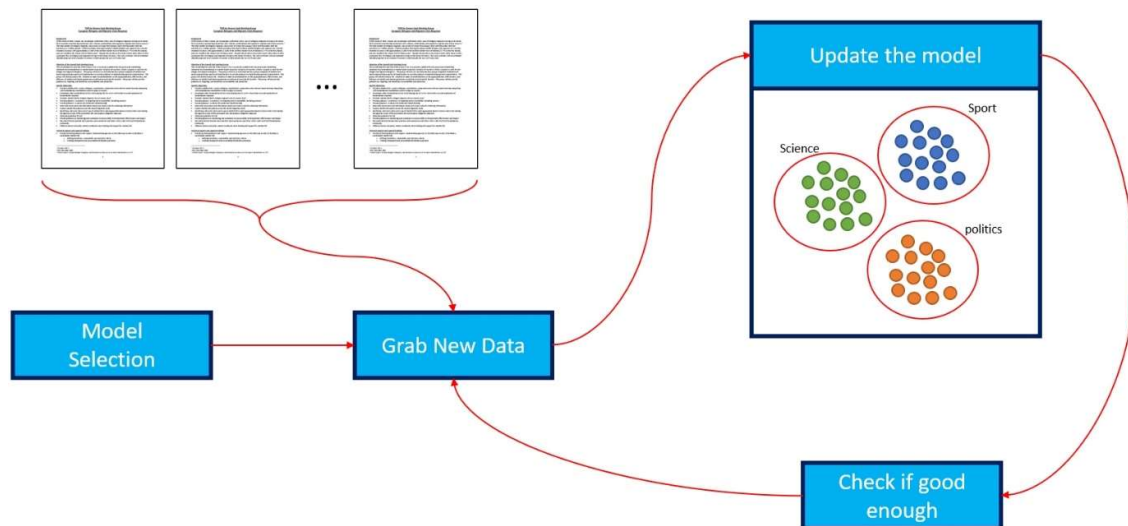
“If you are interested in using machine learning to develop an agent that interacts with an environment and takes actions, then you are probably going to wind up focusing on reinforcement learning.”

الگوریتم های یادگیری تقویتی شامل روش هایی هستند که در آنها عامل هوشمند با محیط تعامل دارد و با توجه به درکی که از شرایط محیط دارد اعمالی را انتخاب میکند و انجام میدهد.

سوال سوم

فرض کنید از شما خواسته شده است که سامانه ای طراحی کنید که عنوان یک خبر را تشخیص دهد، به طوری که مبتنی بر **unsupervised machine learning** کار کند. مراحل طراحی این سامانه را از جمع آوری دیتا تا طراحی مدل همانند مثال ۱.۱ کتاب توضیح دهید.

به وضوح مشخص است ما نمیتوانیم چنین سامانه ای را با کدنویسی تمام حالات این مسئله را حل کرد. پس به سراغ روش های یادگیری ماشین میرویم. در اولین قدم باید مسئله را به صورت دقیقی تعریف کنیم به این معنی که نوع ورودی و خروجی مسئله و همینطور یک خانواده مدل مناسب برای مسئله انتخاب کنیم که به آن **model selection** میگویند و در واقع یکی از مهم ترین قدم ها در حل مسئله است. در مثال ما ورودی مسئله یک متن است و خروجی مدل باید مشخص کننده موضوع متن (ورزشی، اجتماعی، سیاسی و ...) باشد. در قدم بعدی باید به اندازه ی کافی دیتا جمع آوری کنیم تا با استفاده از آنها مدل انتخاب شده را آموزش دهیم. در مثال ما چون باید از یک رویکرد **unsupervised** استفاده کنیم نیازی نیست دیتاست ما برچسب (label) داشته باشد. به همین دلیل خروجی مسئله نمیتواند یک کلاس از موضوعات نوشته باشد و ما فقط میتوانیم با خوشه بندی (**clustering**) متن های شبیه به هم که احتمالا واژگان مشابهی در آنها بیشتر استفاده شده است، متن های جدید کاربر، بعد از آموزش موفقیت آمیز مدل را در دسته مناسب قرار دهیم. حال به شرح مفصل تر چگونگی آموزش مدل میپردازیم. در ابتدا تمام پارامترهای مدل ما رندم هستند و احتمالا مدل هیچ کار مفید و درستی انجام نمیدهد. الگوریتم به مرور به ورود متن های آموزشی یادمیگیرد چگونه میتواند در مدل را مقایسه کند و میزان شباهت آنها را نسبت به متن هایی که قبلا وارد مدل شدند را بسنجد. برای مصورسازی این عملیات تصویر ۱ را رسم کردیم که در پایین مشاهده میکنید.



تصویر ۱ طراحی مدل و آموزش آن به شیوه ی *unsupervised*

سوال سوم

کتابخانه ها و مواردی که در قسمت **installation** در کتاب بیان شده است را نصب کنید و تمرین 2.1.8 و 2.2.5 کتاب را انجام دهید و گزارش کوتاهی برای آنها بنویسید. برای این تمرین مختار هستید از **Pytorch** یا **Mxnet** استفاده کنید.

2.1.8. Exercises

1. Run the code in this section. Change the conditional statement $X == Y$ in this section to $X < Y$ or $X > Y$, and then see what kind of tensor you can get.

در این بخش از کتاب کدهای ابتدایی برای آشنایی و انجام عملیات اولیه با کتابخانه **Pytorch** آمده است. در ابتدا از تعریف متغیرهایی از جنس **tensor** شروع و در ادامه انجام عملیات مختلف مانند اعمال ریاضیاتی پایه و توابع پیشرفته تر نشان داده شده است. همچنین آموزش کار کردن با تمام یا قسمتی از **tensor** های دوبعدی نمایش داده شده است. و اما جواب این قسمت از تمرین این است که وقتی بین دو **tensor** از عملگرهای مقایسه ای استفاده میکنیم نتیجه یک **tensor** دقیقاً با همان ابعاد است و در واقع بین هر جفت عنصر متناظر عملگر مقایسه ای را انجام میدهد و خروجی از جنس **Boolean** یعنی **True** یا **False** برمیگرداند.

2. Replace the two tensors that operate by element in the broadcasting mechanism with other shapes, e.g., 3-dimensional tensors. Is the result the same as expected?

برای پاسخ به این سوال با استفاده از تابع **randint** دو **tensor** سه بعدی با ابعاد ۲ و ۳ و ۴ ایجاد کردیم. سپس آنها را جمع کردیم. در شکل زیر مشاهده میکنید که در مواقعی که ابعاد دو **tensor** دقیقاً یکسان باشد عناصر متناظر در آنها با هم عملیات انجام میدهند.

```
[54]: A = torch.randint(0,10,(2,3,4))
      B = torch.randint(0,10,(2,3,4))
      print('A')
      print(A)
      print('B')
      print(B)
      print('A + B')
      print(A + B)

A
tensor([[[ 8, 7, 1, 3],
          [4, 1, 2, 7],
          [7, 5, 0, 8]],
        [[ 3, 3, 0, 4],
          [1, 8, 0, 6],
          [4, 4, 3, 8]]])

B
tensor([[[ 9, 4, 8, 5],
          [4, 3, 0, 2],
          [5, 5, 0, 5]],
        [[ 4, 7, 5, 2],
          [7, 1, 7, 8],
          [6, 4, 9, 6]]])

A + B
tensor([[[17, 11, 9, 8],
          [ 8, 4, 2, 9],
          [12, 10, 0, 13]],
        [[ 7, 10, 5, 6],
          [ 8, 9, 7, 14],
          [10, 8, 12, 14]]])
```

اما در مواقعی که ابعاد دو تنسور برابر نباشد در حالت داریم. اگر بعد اول یکی از آنها برابر ۱ باشد (برای مثال یک `tensor` با ابعاد $1 \times 2 \times 3$ را در نظر بگیرید) به نحوی آن را کپی سازی میکند تا با `tensor` با ابعاد بزرگتر هم شکل شود و عملیات بدون خطا اجرا میشود اما اگر بعد اول دو `tensor` برابر نباشد و هیچکدام هم یک نباشند در این صورت با خطای `dimension` مواجه میشویم.

2.2.5. Exercises

Create a raw dataset with more rows and columns.

1. Delete the column with the most missing values.

برای این سوال از یک دیتاست کوچک با نام `StoneFlakes` که در `UCI` موجود است استفاده کردیم. پس از تبدیل فرمت فایل از `.data` به `.csv`. فایل را با کتابخانه `pandas` بارگزاری کردیم. سپس تعداد مقادیر `NAN` را در هر ستون شمردیم. ستون شماره ۲ با ۴ مقدار `null` بیشترین تعداد مقادیر `NAN` را داشت و حذف شد.

2. Convert the preprocessed dataset to the tensor format.

پس از حذف ستون دوم، حال با استفاده از میانگین هر ستون، مقادیر `null` را جایگزین میکنیم. و با جدا کردن آخرین ستون (ستون `label`) از `dataframe`، دو `dataframe` جدید بدست میآوریم و در دو `tensor` مختلف ذخیره میکنیم. (کد و دیتاست دو تمرین آخر به همراه این فایل گزارش بارگزاری شدند.)

References

- 1) archive.ics.uci.edu/ml/datasets/StoneFlakes